

REMARKS

Claims 1, 2, 4-9, 17-19, and 21-29 are all the claims presently pending in the application. Claims 3, 10-16, and 20 are canceled. Various claims have been amended to more particularly define the invention.

It is noted that Applicants specifically state that no amendment to any claim herein should be construed as a disclaimer of any interest in or right to an equivalent of any element or feature of the amended claim.

Claims 1, 2, 4-9, 17-19, and 21-29 stand rejected under 35 U.S.C. § 101 as directed toward non-statutory subject matter. Claims 1, 2, 4-9, and 21-29 stand rejected under 35 U.S.C. § 112, second paragraph, as indefinite. Claims 1, 4, 17-19, and 21-29 stand rejected under 35 U.S.C. § 102(e) as anticipated by U.S. Patent No. 7,031,994 to Lao et al., and claims 2 and 5-9 stand rejected under 35 U.S.C. § 103(a) as unpatentable over Lao.

These rejections are respectfully traversed in the following discussion.

I. THE CLAIMED INVENTION

The claimed invention, as described, for example, in independent claim 1, is directed to a computer including a processor, a memory system, a co-processing unit, and a plurality of data registers for data exchange with the co-processing unit. The computer is controlled to implement a method of increasing efficiency in executing a matrix operation that uses matrix data in a standard format, the standard format comprising one of a column major format and a row major format. The method comprises, for matrix data stored in the standard format in the memory system, wherein the matrix data comprises data of any of a complete matrix, a complete submatrix, or a part of a matrix or submatrix, using the processor to separate the matrix data into blocks of data, each block having a size p-by-q.

The processor then rearranges and places in the memory system of the computer, for retrieval in a repetitive manner for executing the matrix operation, the blocks of data to be contiguous blocks of contiguous data such that the matrix data is represented in a nonstandard format that permits the matrix data to be moved from the memory system into a position for performing the matrix operation more quickly than if the matrix data had been moved as stored in the standard format.

In another aspect of the present invention described by independent claims 26 and 29, the present invention permits a hardware/instruction set deficiency to be overcome by using software instructions only. The software instructions would actually seem to be performing two errors relative to data for the intended operation, but the two errors together are designed to overcome the computer deficiency without having to redesign the machine and by using conventional compilers. In the exemplary embodiment, the deficiency involves the interface with the FPU, the co-processing unit that actually executes the matrix operation described, although the concept is clearly more general.

The present invention is actually only one of several ways used by the inventors to improve efficiency of matrix processing on the assignee's BlueGeneL computer. As explained beginning at line 19 on page 3, the present invention involves a method that improves efficiency and/or overcomes a hardware/instruction set deficiency without going through the expense of a re-design of computer hardware or instructions.

Conventional wisdom would have corrected the deficiency by redesigning the chip in the computer.

The claimed invention, on the other hand, provides a software solution that can be accomplished using conventional compilers and conventional assemblers. Applicants submit that this method is clearly non-obvious, since it involves intentionally executing two errors relative to the intended matrix operation on standard matrix data. That is, the present invention teaches a preliminary processing that converts the matrix data into non-standard format and a new processing added to the matrix operation so that the overall result will be correct for the matrix operation. In the example described in the disclosure, the second error involved loading data into the FPU registers using a non-standard loading instruction that criss-crosses the loading.

However, it is noted that the feature described in independent claim 1 also has utility separate from overcoming a design deficiency, since the standard matrix format using row/column major format is typically a disadvantage for at least one of the three matrices involved in, for example, a matrix multiplication process.

II. THE 35 USC §101 REJECTIONS

Claims 1-20 stand rejected under 35 U.S.C. §101 as allegedly directed to non-statutory subject matter. Applicants respectfully disagree, for the following reasons.

The Examiner reasons: *“It is clear from the claims that the invention merely involves data manipulations for rearranging data of a matrix. The claimed invention does not transform an article or physical object to a different state or thing, and the result produced by the claimed invention is a mere set of data being rearranged from an original data matrix and does not have a real world value, and thus [does] not [provide] a useful, concrete and tangible[result]. Therefore, the claimed invention is directed to non-statutory subject matter for failing to accomplish a practical [application].”*

Applicants respectfully disagree with the above-recited evaluation. The present invention arose from a technical problem at the FPU interface. Newer FPU units attached to computers do not have adequate hardware/software provisions for efficiently exchanging data for matrix processing. The present inventors have recognized this interface problem and have provided a solution to this hardware/software problem that involves a rearrangement of data that will ultimately be repetitively moved into the FPU for the processing, thereby overcoming the inefficiency at the FPU interface without having to modify the hardware of the FPU or computer.

Thus, contrary to the Examiner’s characterization, the present invention as a whole inherently provides a practical application. Therefore, even if the Examiner wishes to characterize the present invention as “merely” rearranging the matrix data, such description is clearly incorrect in its characterization that the invention is not accomplishing a practical application, since the increase in efficiency and the overcoming of a hardware/software interface deficiency using a rearrangement of data *inherently* accomplishes a practical application.

As explained at line 19 on page 3 through line 4 on page 4, Applicants observed during development of the BlueGeneL computer that this computer architectures failed to have an operation specifically addressed to some stages of matrix processing and that one novel method of overcoming such deficiencies in a computer would be to store the matrix data in a non-standard format (the “register block data format”, see line 18 on page 7) that actually changes order of the information content of the matrix (see “pseudo-matrix” nomenclature in line 20 on page 7) and constitutes a first “error” in the matrix processing. As explained at line 20 on page 7 through line 3 on page 8, a correcting “error” is then intentionally executed that places the information into the computer architecture and instruction set in a manner that “corrects” the first error so that the matrix processing efficiency is improved in a manner that a standard compiler can execute and without having

to redesign the computer architecture, components, and/or instruction set.

In the exemplary embodiment discussed in the present invention, the data of the matrix is intentionally placed in “improper” locations as a preliminary processing, thereby changing the order of the information content of the matrix data if this data were to be used as stored. The second “error” in the example of the present invention is defined in claim 2, wherein matrix data from storage is loaded into FPUs in a non-standard manner that corrects for the dislocation of matrix data, thereby actually increasing efficiency in the matrix processing, while, in the exemplary embodiment, also overcoming an underlying deficiency in the machine architecture and/or instruction set relative to the matrix processing.

Applicants submit, therefore, that this rejection directed to non-statutory subject matter clearly is based upon a misunderstanding of the significance of the claimed invention, since this invention clearly has “real-world” application, by increasing efficiency of processing on a computer, and in the exemplary embodiment, being a part of a mechanism that overcomes one or more inherent deficiencies relative to the matrix processing in that machine.

The present invention achieves this increase in efficiency by adding preliminary data processing steps to reformat the data for at least one of the matrices involved in the matrix operation. That is, rather than changing the machine architecture, modifying/ redesigning the machine components, or developing a new machine instruction, the claimed invention describes a method of making corrections for the hardware deficiencies of the machine by converting the matrix data into a format that is different relative to the matrix data stored in standard format.

A correcting “error”, which in the exemplary embodiment, comprises rewriting the matrix operation code to accommodate the nonstandard format by, for example, loading the pseudo matrix data into the FPU register set using optimal loading instructions that can be executed by the processor. Together, these two errors will place the matrix data into the correct location for processing in the FPU, while allowing the data to have been moved faster than if the data had been moved in standard column/row major format for that matrix data.

The initial penalty of preprocessing the data to place it into the non-standard format is more than compensated overall because this data will be retrieved repeatedly for the linear algebra processing by the FPU. The initial “one-time” preprocessing pays off because of the repetitive retrieval of the data from cache into the FPU data registers.

Moreover, the predetermined non standard loading, a software processing, permits the

data to be loaded optimally in the FPU data registers, thereby overcoming a non-optimal hardware design at the FPU interface, using software instructions rather than changing the machine hardware or instruction set.

Therefore, Applicants submit that the method claims, when properly understood, clearly satisfy the “useful, tangible, and concrete” standard required to allow computer-related processes to be statutory subject matter.

Relative to claims 17-19, Applicants again submit that these claims are clearly worded to be directed toward “A computer-readable medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus” and, therefore, not at all directed to a “... signal carrier which is non-statutory subject matter”, as characterized by the rejection. Rather, these claims are directed toward Beauregard claims and are clearly patentable (see *In re Beauregard*, 53 F.3d 1583 (Fed. Cir., 1995)), as well as US Patent 5,710,578 to Beauregard et al.

In view of the foregoing, the Examiner is respectfully requested to reconsider and withdraw these rejections for claims 1-9 and 17-19.

Relative to claims 21-29, the claimed invention, as a whole, is directed to either improving efficiency of executing a process on a computer or overcoming a hardware/instruction set deficiency of a specific machine configuration by using software instructions that can be handled by conventional compilers. Both these aspects are “real world results” and, therefore, these new claims are clearly directed toward statutory subject matter.

Therefore, the Examiner is respectfully requested to reconsider and withdraw this rejection based on 35 USC §101.

III. THE REJECTION UNDER 35 U.S.C. 112, SECOND PARAGRAPH

The Examiner alleges that claims 2 and 5-9 are indefinite because claim 2 recites “... a format of data ... comprises variations of an optimal floating point loading instruction.” Applicants respectfully disagree, since claim language is required to be interpreted in view of the specification as understood by one having ordinary skill in the art. Moreover, dependent claim 5 provides an example of an exemplary variation described in the specification.

Relative to claim 28 and 29, Applicants believe the changes above appropriately address the Examiner’s concerns.

Therefore, the Examiner is requested to reconsider and withdraw this rejection.

IV. THE PRIOR ART REJECTION

The Examiner alleges that Lao teaches the claimed invention. Applicant submits, however, that there are elements of the claimed invention which are neither taught nor suggested by Lao.

Lao discloses a method to transpose a matrix, as described in the Abstract. Transposition of matrix data is a common matrix procedure that is characterized as preserving the standard format of the matrix. In contrast, the register data format of the present invention places the matrix data into nonstandard format wherein the matrix data is no longer in row major or column major format. The inventors have recognized that storing matrix data in this nonstandard format can speed up the movement of matrix data to the FPU, as well as potentially being a part of a mechanism that overcomes a deficiency in a specific computer architecture/instruction set relative to matrix processing.

The present invention is not simple matrix transposition, as the Examiner seems to consider in comparing it with Lao. This can be seen from Figure 6 of the present application, wherein matrix A^T 602 shows the simple transpose of matrix A 601. As explained beginning at line 1 on page 18, the pseudo matrix A 605 of the present invention clearly differs from the transpose. To achieve the transpose 601, the present invention performs the crisscross loading into the FPU registers, as defined by claim 5, but even this “transposition” is only for the specific block being loaded by the crisscross loading. That is, there is no complete transposed matrix in the present invention similar to that taught in Lao. Thus, contrary to the Examiner’s position, the present invention does not provide storage of the matrix transpose as done in Lao, and Lao does not provide the blocks of contiguous data defined in the independent claims.

Although there are many permutations of the matrix data that are possible, the present invention teaches that at least one of these many permutations will allow the matrix data for at least one of the matrices involved in a matrix multiply processing to be rearranged in memory so that the matrix data can for that matrix can be retrieved for processing by the FPU as contiguous data for more efficient transport and processing, albeit, in non standard matrix format. As shown in Figure 6, this pseudo matrix 605 does not correspond to the transpose of the matrix data. Rather, it is the permutation (of the many possible permutations) that the inventors have recognized could be utilized to overcome the deficiency at the FPU register

interface.

This concept of the present invention arises because the standard column/row major format for matrix data will be a disadvantage for retrieving data of at least one of the three matrices involved in matrix multiplication. Additionally, as noted, the present invention addresses a hardware design deficiency at the FPU interface.

That is, as a non-limiting example of a matrix operation, the discussion in the specification addresses specifically matrix multiplication. However, the concepts of the present invention are equally applicable to many other matrix operations and could be extended to more general processing, particularly relative to the aspect of the present invention of overcoming one or more design or interface deficiencies by using software instructions that can be executed with a standard compiler.

Therefore, in contrast to the routine matrix transposition processing in Lao, in one exemplary embodiment, the present invention is directed to the entirely different problem of overcoming a shortcoming in a chip of a computer wherein either the computer architecture or the computer instruction set is deficient, or at least inefficient, relative to the intended matrix processing. The solution proffered by the present invention involves using software that can be implemented by conventional compilers, rather than re-designing the computer chip and/or instruction set.

As defined by the independent claims, the solution involves rearranging the matrix data into a specific nonstandard format such that the matrix data is no longer stored in the standard row major/column major format.

In the exemplary embodiment described in the specification, this rearranging of the matrix data constitutes a first error relative to “normal” matrix processing. In a subsequent step, a second error relative to the intended matrix processing is executed, which in the exemplary embodiment, involves loading the data into the FPU in a cross-loading pattern.

Together, the two errors of the preliminary processing combine in a manner that corrects the shortcoming of the computer chip.

More specifically, during development of the BlueGeneL computer, the present inventors recognized that the standard memory layout of matrices (e.g., column major or row major order) could to be changed to a layout (e.g., Register Block Data Format, RBDF) prior to the intended processing by Dense Linear Algebra (DLA) routines. These subsequent DLA routines have a common property in that they reuse their data repeatedly while it resides in L1 cache and/or L0 cache. The L0 cache is intended, in the present invention, as referring to

the Floating Point Register File (FPRF) of a Floating Point Unit (FPU).

Recently, computer architectures have added special fast floating point units as attachments to their basic processing units. So, to be able to use these units effectively, the present inventors recognized that it is no longer sufficient just to bring data into the L1 cache, since the data should enter L1 and L0 in an optimal seamless manner relative to the new FPUs. The novel RBDF of the present invention has this optimal seamless property whereas conventional column major order or row major order does not.

Hence, as described exemplarily in independent claim 1, the present invention teaches that it pays to reorder standard matrix data layouts, such as CM order, into RBDF before invoking the subsequent DLA routines, as a preliminary processing. Of course, the subsequent DLA routines “know” that their inputs are in RBDF and not standard CM order, as described in dependent claims 2 and 23.

Therefore, a rationale for the present invention now becomes apparent in view of this explanation. Any gain that occurs from using RBDF, instead of CM order, gets multiplied by the repetitive factor inherent in a given subsequent DLA routine that will be using the novel RBDF format of the present invention. Hence, the one-time cost of initially converting the standard CM order into RBDF is quickly paid for. Overall, there is a substantial performance gain in using RBDF, so that the conversion described in the independent claims increases efficiency of the intended matrix operation, while also overcoming the deficiency inherent in newer FPUs by using a software remedy.

Thus, the present invention is not related to the transposition technique described in Lao, since the present invention stores the matrix data in a non-standard format as a preliminary process to the intended matrix operation. A subsequent second error compensates for the incorrect information content. The matrix transposition of Lao is not a preliminary processing, since this transposition processing in Lao is actually the desired matrix operation itself.

Stated slightly differently, the present invention can be viewed as providing a method and structure for high performance processing of linear algebra routines using register block data format (RBDF) routines, wherein conversion into RBDF is a preliminary processing. The register block data format can then be subsequently used by many Dense Linear Algebra (DLA) algorithms. In contrast, Lao, at most, obtains instances of special cases of register block format but subsequently destroys these instances by returning its final output in standard column major (CM) order, and these instances are, therefore, not preliminary

processing.

Lao addresses the process of transposing a matrix as the desired matrix processing. In this process, the matrix data starts in standard CM order and output is always returned to standard CM order. In contrast, one aspect of the present invention involves changing the standard CM order input of a matrix A into a new data format RBDF (register block data format) that is not the same as the matrix transpose (reference Figure 6 of the present application) that Lao accomplishes.

In paragraph 7 on page 4 of the Office Action, the Examiner refers to Figure 6 of Lao. This figure shows a block diagram of a computer performing the transposition process shown in the flowchart of Figure 5. These figures refer to a special case of a rectangular matrix where M, the number of rows, is related to N, the number of columns, by the formula $N=kM$. The Examiner alleges that "..., each said block having a size 2-by-2;"

However, Applicants submit that Lao fails to discuss 2-by-2 blocks in Figure 6 and that the Examiner seems to conclude that this 2-by-2 terminology related to column 16, lines 27-59, where lines 49-53 discuss a checkerboard loading pattern, is similarly related to Figure 6. Applicants submit that the discussion in lines 27-59 of column 16 is related to a specific example for transposing a matrix A with M=6 rows and N=2 columns.

Applicants submit that it is well known in the art that to transpose a 2-by-2 matrix, one exchanges the elements in the (2,1) and (1,2) positions. This fact is how "checkerboarding" enters into Lao. Lao is forced to use checkerboarding for 2-by-2 submatrices if it is desired to transpose a 2-by-2 submatrix of a larger matrix A.

In contrast, the present invention sometimes will perform a 2-by-2 transposition, perhaps even by using standard out-of-place matrix transposition techniques, to produce the RBDF. However, Lao's purpose is that of a matrix in-place transposition by a faster technique. The present invention has an exemplary purpose of producing a new data format RBDF, for subsequent repetitive use by DLA routines, an entirely different purpose from that of the transposition processing in Lao.

Additionally, Applicants submit that the description in lines 27-59 of column 16 relates to one step of an algorithm described in Figures 7 and 8, along with line 62 of column 16 to line 57 of column 18. An important point is that this is an intermediate step of Figures 7 and 8 and that this instantaneous data layout is subsequently destroyed and returned to standard CM order.

Thus, the present invention clearly differs from Lao, wherein the matrix operation is

simple matrix transposition, in that the rearranging of matrix data is a preliminary step to the matrix operation that results in the matrix data being stored for retrieval in a non-standard matrix format. The “matrix operation” in the present invention exemplarily involves a DLA processing.

Hence, turning to the clear language of the claims, in Lao there is no teaching of: “... separating said matrix data into blocks of data, each said block having a size p-by-q; and rearranging and placing in said memory system of said computer, for retrieval in a repetitive manner for executing said matrix operation, said blocks of data to be contiguous blocks of contiguous data such that said matrix data is represented in a nonstandard format that permits said matrix data to be moved from said memory system into a position for performing said matrix operation more quickly than if said matrix data had been moved as stored in said standard format”, as required by independent claim 1. The remaining rejected independent claims have similar wording and/or wording that clearly distinguishes from the standard format processing of Lao, since Lao is directed toward the process of transposing matrix data rather than overcoming a deficiency of an FPU interface.

In yet another evaluation of Lao relative to the present invention, Lao has ten figures and five cases, two figure per case, as follows:

Case 1 (Fig. 1, 2): is the main claim of Lao. Here the arrangements are standard. Any non-standard movement has to do with permutation movement which is not adequately described in Lao. Restriction: $LDA = M$ and q divides N .

Case 2 (Fig. 3, 4): $M = N$. The prior art works just as well here which Lao is using but not saying so.

Case 3 (Fig. 5, 6): $N = k * M$. Special case, like case 1. Restrictions: $LDA = M$ and $N = k * M$.

Case 4 (Fig. 7, 8): $M = k * N$. Here a data rearrangement technique is described in step 702. As in case 1, this involves permutation movement which is not adequately described in Lao. At this point, one has k M by M matrices and the case 2 is applied k times.

It is noted that, in the very special case when $M = 2$, Lao’s data rearrangement is similar to the effect on portions of data of the present invention after the crisscross loading into the FPU data registers. However, even in this special case, Lao clearly has no corresponding storage of data to be used retrieved repetitively, let alone retrieval to then be subjected to a crisscross loading pattern.

Case 5 (Fig. 9, 10): $M = k \cdot M$ & $N = k \cdot n$. Requires a buffer of size $k \cdot (m \cdot n)$. Then standard copies are used with buffer and storage twice. Next, case 5 uses standard arrangements. Finally, standard block swaps are used. Restrictions: $LDA = M$, $M = k \cdot m$, $N = k \cdot n$.

In summary, the only case in Lao that is similar to the present invention is case 4 and then only when $M = 2$. However, as indicated above, even in this similar case, there is no correspondence with the present invention until the crisscross loading into the FPU data registers and Lao has no corresponding concept of using a special, new data format (e.g., RBDF) for repetitive retrieval for processing.

Relative to the Examiner's position for rejecting claims that specifically identify the FPU, Applicants respectfully submit that this rejection fails to meet the initial burden of a *prima facie* rejection. As Applicants have explained, the present invention arose due to the inventors' recognition that the newer FPUs had a deficiency at the data exchange interface. Rather than redesign the hardware, the present invention provides a solution that a standard compiler can implement.

There is no suggestion in Lao or any other prior art currently of record that this FPU deficiency is known, let alone a solution along the lines of that proposed by the present invention. Therefore, to the extent that the Examiner is considered to have invoked official notice, Applicants respectfully request that the Examiner provide a properly-combinable reference on the record that addresses the plain meaning of the claim language prior to proceeding to Appeal.

It is further noted that the criss-cross loading pattern used in the present invention is used as a non-standard loading instruction of data into the FPU data registers. There is reference currently of record (including Lao) that suggests such criss-cross data loading into an FPU.

Therefore, Applicant submits that there are elements of the claimed invention that are not taught or suggest by Lao, and the Examiner is respectfully requested to withdraw this rejection.

V. FORMAL MATTERS AND CONCLUSION

In view of the foregoing, Applicants submit that claims 1, 2, 4-9, 17-19, and 21-29, all the claims presently pending in the application, are patentably distinct over the prior art of

Serial No. 10/671,888
Docket No. YOR920030169US1 (YOR.463)

record and are in condition for allowance. The Examiner is respectfully requested to pass the above application to issue at the earliest possible time.

Should the Examiner find the application to be other than in condition for allowance, the Examiner is requested to contact the undersigned at the local telephone number listed below to discuss any other changes deemed necessary in a telephonic or personal interview.

The Commissioner is hereby authorized to charge any deficiency in fees or to credit any overpayment in fees to Assignee's Deposit Account No. 50-0510.

Respectfully Submitted,



Date: June 19, 2007

Frederick E. Cooperrider
Registration No. 36,769

McGinn Intellectual Property Law Group, PLLC
8321 Old Courthouse Road, Suite 200
Vienna, VA 22182-3817
(703) 761-4100
Customer No. 21254

CERTIFICATION OF TRANSMISSION

I certify that I transmitted electronically, via EFS, this Amendment under 37 CFR §1.116 to the USPTO on June 19, 2007.



Frederick E. Cooperrider
Reg. No. 36,769